An Optimized In-Network Aggregation Scheme for Data Collection in Periodic Sensor Networks

Jacques Bahi, Abdallah Makhoul, Maguy Medlej

An Optimized In-Network Aggregation Scheme for Data Collection in Periodic Sensor Networks

Jacques M. Bahi, Abdallah Makhoul, and Maguy Medlej

FEMTO-ST Laboratory, DISC departement
University of Franche-Comté
Rue Engel-Gros, 90016 Belfort, France
firstname.lastname@univ-fcomte.fr

Abstract. In-network data aggregation is considered an effective technique for conserving energy communication in wireless sensor networks. It consists in eliminating the inherent redundancy in raw data collected from the sensor nodes. Prior works on data aggregation protocols have focused on the measurement data redundancy. In this paper, our goal in addition of reducing measures redundancy is to identify near duplicate nodes that generate similar data sets. We consider a tree based bi-level periodic data aggregation approach implemented on the source node and on the aggregator levels. We investigate the problem of finding all pairs of nodes generating similar data sets such that similarity between each pair of sets is above a threshold t. We propose a new frequency filtering approach and several optimizations using sets similarity functions to solve this problem. To evaluate the performance of the proposed filtering method, experiments on real sensor data have been conducted. The obtained results show that our approach offers significant data reduction by eliminating in network redundancy and outperforms existing filtering techniques.

1 Introduction

Data collection from sensor networks can be made on demand or by data streaming. The first category is done by bi-directional dialogs between the sensor nodes and the base station. A request for data is sent from the end user via the sink to the sensor nodes which, in return, send back the data to the user via multi hop communications. On the other side, in data streaming, data flows primarily from the sensor node to the sink. In this category we distinguish the periodic sampling and the event driven data models. In this paper we are interested in ”periodic sampling” data model in sensor networks, where the acquisition of sensor data from a number of remote sensor nodes are forwarded to the gateway on a periodic basis. This data model is appropriate for applications where certain conditions or processes need to be monitored constantly, such as the temperature in a conditioned space or pressure in a process pipeline. There are couple of important design considerations associated with the periodic sampling data model. The most critical design issue is the phase relation among multiple sensor nodes. If two neighbor nodes operate with identical or similar sampling rates, redundant packets from the two nodes are likely to happen repeatedly. It is essential for sensor networks to be able to detect and clean redundant transfered data from the nodes to

the sink. In-network data aggregation has been proven as an effective technique for eliminating redundancy and forwarding only the extracted information from the raw data. Furthermore, by doing so data aggregation can often reduce the communication cost and extend the whole network lifetime.

In this paper we present a hierarchical multilevel data aggregation scheme aiming to optimize the volume of data transmitted thus saving energy consumption and reducing bandwidth on the network level. A first level in-sensor process is done by the nodes themselves. Instead of sending each sensor node's raw data to a base station, the data is cleaned periodically by the sensor node itself before sending it to an aggregator node for a second level of aggregation. At this level, we are interested in exploring a new part of the filtering aggregation problem, by focusing on identifying the similarity between data sets generated by neighboring nodes and sent to the same aggregator. Our objective is to identify similarities between near sensor nodes, and integrate their captured data into one record while preserving information integrity.

In this paper, we provide a new prefix filtering method to study the sets similarity in sensor networks. We propose frequency filtering optimization techniques, which exploits the ordering of measurements according to their frequencies. A frequency of a measure is defined by the number of occurrences of this measure in the set defined at the first aggregation level. Furthermore, we provide a new optimization method for early termination of sets similarity computing. To evaluate our approach we conducted extensive experimental study using real data measurements. The obtained results compared to the existing algorithms show the effectiveness of our method which significantly reduces the number of duplicate data.

The rest of the paper is organized as follows, Section 2 gives an overview on related works reported on data aggregation in sensor networks. Section 3 describes our periodic data aggregation scheme. The local aggregation level is presented in section 4. Review on similarity functions and our proposed frequency filtering techniques are presented in Section 5. Experimental results are given in Section 6. Section 7 concludes the paper with some directions to a future work.

2 Previous Data Aggregation Work

Data aggregation in wireless sensor networks has been well studied in recent years [1] [2] [3]. It means computing and transmitting partially aggregated data to the end user rather than transmitting raw data in networks to reduce the energy consumption [4]. There are vast amount of extant works on in-network data aggregation in the literature.

Some of the methods reported recently are query based methods [5] [6]. A query is generated at the sink and then broadcasted through the network. Some nodes just process the query, while others propagate it, receive partial results, aggregate results, and send them back to the sink. Various algorithmic techniques have been proposed to allow efficient aggregation without increasing the message size [7].

Some works, such as [8] [9] [10], use the clustering methods for aggregating data packets in each cluster separately. Among these methods, the LEACH protocol [11] [12]. In [9], the authors propose a self-organizing method for aggregating data based on the architecture CODA (Cluster-based self-Organizing Data Aggregation), based on the

Kohonen Self-Organizing Map to aggregate sensor data in cluster. In a first step before deployment, the nodes are trained to have the ability to classify the sensor data. Thus, it increases the quality of data and reduces data traffic as well as energy-conserving. An adaptive data aggregation (ADA) scheme for clustered sensor networks has been proposed in [10]. In this scheme, a time based as well as spatial aggregation degrees are introduced. They are controlled by the reporting frequency at sensor nodes and by the aggregation ratio at cluster heads (CHs) respectively. The function of the ADA scheme is mainly performed at the sink node, with a little function at CHs and sensor nodes.

In a tree based network as our presented work, sensor nodes are organized into a tree where data aggregation is performed at aggregators along the tree to arrive to the sink. Tree based data aggregation approaches are suitable for in-network data aggregation. The authors in [13] [14], have proposed Tree on DAG (ToD) for data aggregation, a semistructured approach that uses Dynamic Forwarding on an implicitly constructed structure composed of multiple shortest path trees to support network scalability. The key principle behind ToD was that adjacent nodes in a graph will have low stretch in one of these trees in ToD, thus resulting in early aggregation of packets.

In our previous work [3], we have shown that existing prefix filtering methods are very complex and not suitable for sensor networks and we proposed a heuristic based on the frequency ordering. In this paper, we propose two optimization techniques based on frequency filtering extention which can be integrated with our previous prefix method [3] to find similar data sets efficiently. Furthermore we provide a new and faster technique for sets similarity computation.

3 Periodic Data Aggregation

Due to resource restricted sensor nodes, it is important to minimize the amount of data transmission among sensor networks so that the average network lifetime and the overall bandwidth utilization are improved. To reduce the amount of sending data, an aggregation approach can be applied along the path from sensors to the sink. Sensor nodes collect information from the region of interest and send it to aggregators. Each aggregator then condenses the data prior to sending it on.

Our data aggregation method works in two phases, the first one at the nodes level, which we call local aggregation and the second at the aggregators level. At each period p each node sends its aggregated data set to its proper aggregator which subsequently aggregates all data sets coming from different sensor nodes and sends them to the sink.

4 Local aggregation

In periodic sensor networks, we consider that each sensor node i at each slot s takes a new measurement y_{is}. Then node i forms a new set of captured measurements M_i with period p, and sends it to the aggregator. It is likely that a sensor node takes the same (or very similar) measurements several times especially when s is too short. In this phase of aggregation, we are interested in identifying locally duplicate data measurements in order to reduce the size of the set M_i. Therefore, to identify the similarity between two measures, we provide the two following definitions:

Definition 1 ($link$ **function).** *We define the* $link$ *function between two measurements as:*

$$link(y_{is_1}, y_{is_2}) = \begin{cases} 1 & \textit{if} \quad \|y_{is_1} - y_{is_2}\| \leq \delta, \\ 0 & \textit{otherwise}. \end{cases}$$

where δ is a threshold determined by the application. Furthermore, two measures are similar if and only if their $link$ function is equal to 1.

Definition 2 (Measure's frequency). *The frequency of a measurement* y_{is} *is defined as the number of the subsequent occurrence of the same or similar (according to the* $link$ *function) measurements in the same set. It is represented by* $f(y_{is})$.

Using the notations defined above the local aggregation algorithm is done as follows [3]. For each new sensed measurement (at each slot), a sensor node i searches for the similar measure already captured. If a similar measurement is found, it deletes the new one while incrementing the corresponding frequency by 1, else it adds the new measure to the set and initialize its frequency to 1. At the end of the period p, each node i will possess a local aggregated set M_i and send it to its aggregator.

5 Duplicate data sets aggregation

At this level of aggregation, each aggregator has received k sets of measurements and their frequencies. The idea here is to identify all pairs of sets whose similarities are above a given threshold t. For this reason we use a similarity function which measures the degree of similarity between the two sets and returns a value in $[0, 1]$. A higher similarity value indicates that the sets are more similar. Thus we can treat pairs of sets with high similarity value as duplicates and reduce the size of the final data set that will be sent to the sink.

5.1 Similarity Functions

A variety of similarity functions have been used in the literature such as overlap threshold, Jaccard similarity and Cosine similarity [15–17]. We denote $|M_i|$ as the number of elements (measures) in the set M_i. The following functions can be used to measure the similarity between two sets of measurements M_i and M_j :

Overlap similarity: $O(M_i, M_j) = |M_i \cap M_j|$
Jaccard similarity: $J(M_i, M_j) = \frac{|M_i \cap M_j|}{|M_i \cup M_j|}$
Cosine similarity: $C(M_i, M_j) = \frac{|M_i \cap M_j|}{\sqrt{|M_i| \times |M_j|}}$
Dice similarity: $D(M_i, M_j) = \frac{2 \times |M_i \cap M_j|}{|M_i| + |M_j|}$

All these functions are commutative and can be transformed to the Overlap similarity easily. For instance, we can present the Jaccard similarity function as follows:

$$J(M_i, M_j) = \frac{O(M_i, M_j)}{|M_i| + |M_j| - O(M_i, M_j)}$$

In our approach, we will focus on the Jaccard similarity. It is one of the most widely accepted function because it can support many other similarity functions [16]. In our application, two given sets M_i and M_j are considered similar if and only if:

$$J(M_i, M_j) \geq t$$

where t is a threshold given by the application itself. This equation can be transformed as:

$$J(M_i, M_j) \geq t \Leftrightarrow O(M_i, M_j) \geq \alpha \tag{1}$$

where, $\alpha = \frac{t}{1+t}.(|M_i| + |M_j|)$.

In order to study the similarity functions for data aggregation in sensor networks, we define a new function for overlapping "$\cap_s$" between two sets of measurements as follows:

Definition 3 (Overlap function). *Consider two sets of measurements M_1 and M_2, then we define:*

$M_1 \cap_s M_2 = \{(y_1, y_2) \in M1 \times M2$ *such that* $link(y_1, y_2) = 1\}$*; and* $O_s(M_1, M_2) = |M_1 \cap_s M_2|$.

To evaluate the similarity between two sets we obtain:

$$J(M_i, M_j) \geq t \Leftrightarrow |M_i \cap_s M_j| \geq \alpha = \frac{t}{1+t}.(|M_i| + |M_j|) \tag{2}$$

5.2 Sets similarity computation

In this section we provide techniques for computing the similarity between the received sets. A naïve solution to find all similar sets is to enumerate and compare every pair of sets. This method is obviously prohibitively expensive for large data sets (such the case of sensor networks), as total number of comparison is $O(n^2)$.

To reduce the number of comparisons between sets a prefix filtering method has been proposed. Several approaches for traditional similarity join between sets are based on the prefix filtering principle [15] [17] [3]. This method is based on the intuition that if all sets of measures are sorted by a global ordering, some fragments of them must share several common tokens with each other in order to meet the threshold similarity. An inverted index maps a given measurement m to a list of identifiers of sets that contain m_i such that $link(m_i, m) = 1$. After inverted indices for all measures in the set are built, we can scan each one, probe the indices using every measure in the set M, and obtain a set of candidates; merging these candidates together gives us their actual overlap with the current set M; final results can be extracted by removing sets whose overlap with M is less than $\lceil \frac{t}{1+t}.(|M_i| + |M_j|) \rceil$(Equation 1).

This intuition is formalized by the following $Lemma$ inspired from [17]:

Lemma 1. *Consider two sets of sensor measures M_i and M_j, such that their elements are ordered by a global defined ordering. Let the p-$prefix$ be the first p elements of M_i. If $|M_i \cap_s M_j| \geq \alpha$, then the $(|M_i| - \alpha + 1)$-$prefix$ of M_i and the $(|M_j| - \alpha + 1)$-$prefix$ of M_j must share at least one element.*

Proof. Lemma 1 can be proven similarly to the lemma of page 6 in [17].

To ensure the prefix filtering based approach does not miss any similarity set result, as shown in Lemma 1 we need a prefix of length $|M_i| - \lceil t.|M_i| \rceil + 1$ for every set M_i [3]. The algorithm for finding similarity sets based on prefix filtering technique is given in Algorithm 1. It takes as input a collection of datasets coming from different sensor nodes already sorted according to a defined ordering. It scans sequentially each set M_i, selects the candidates that intersects with its prefix. Afterwards, M_i and all its candidates will be verified against the jaccard similarity threshold to finally return the set of correct similar measurements sets.

Algorithm 1 Prefix-filtering based algorithm.

Require: Set of measures' sets $M = \{M_1, M_2...M_n\}$, and a threshold t.
Ensure: All pairs of sets (M_i, M_j), such that $J(M_i, M_j) \geq t$.

$S \leftarrow \emptyset$
$I_i \leftarrow \emptyset$ ($1 \leq i \leq$ total number of measures)
for each set $M_i \in M$ **do**
  $p \leftarrow |M_i| - \lceil t \times |M_i| \rceil + 1$
  $X \leftarrow$ empty map from set id to int
  for $k \leftarrow 1$ to p **do**
    $w \leftarrow M_i[k]$
    if (I_{w_s} exists such that $link(w, w_s) = 1$) **then**
      for each Measurement $(M_j[l]), f(M_j[l]) \in I_{w_s}$ **do**
        $X[M_j] \leftarrow X[M_j] + 1$
      end for
      $I_{w_s} \leftarrow I_{w_s} \cup \{M_i\}$
    else
      create I_w
      $I_w \leftarrow I_w \cup \{M_i\}$
    end if
  end for
  for each M_j such that $X[M_j] > 0$ **do**
    if $O_s(M_i, M_j) \geq \alpha$ **then**
      ($S \leftarrow \{(M_i, M_j)\}$)
    end if
  end for
end for
return S

Prefix filtering algorithm helps prune out unfeasible sets of measures, however, in practice the number of non-similar sets surviving after this technique is still quadratic growth [18]. Following the prefix filtering, many optimization methods [18] [19] were proposed to prune out further the unfeasible non-similar sets. A trade-off of these prefix filtering optimizations is that usually require more computational efforts which is unsuitable by heavy resources sensor networks. In our approach, we provide some opti-

mizations for prefix filtering techniques based on measures frequency while taking into account this trade-off.

5.3 Frequency filtering approach

In this section, we present our frequency filtering method based on prefix extension. We begin by introducing some definitions and notations which will be the basis of what follows. In periodic sensor networks, two data sets are similar if their measurements overlap with each other, and especially the ones having *higher frequencies values*.

Definition 4 (Ordering $\mathcal{O}$). *We define an ordering $\mathcal{O}$ which arranges the measurements of a given set by the decreasing order of their frequencies.*

For two similar measures m_i and m_j such that $link(m_i, m_j) = 1$, we denote $f_{min}(m_i, m_j) = Min(f(m_i), f(m_j))$ the minimum value of the frequency of these measures.

Definition 5 ($f_s(M_i, M_j)$). *Consider two sets of measures M_i and M_j, we define* $f_s(M_i, M_j) = \sum_{k=1}^{O_s(M_i,M_j)}(f_{min}((m_i, m_j) \in M_i \cap_s M_j))$.

In this paper, we consider that all sensor nodes operate with the same sampling rate, and every node captures τ measures with each period p. Thus we can deduce that for every received set M_i from node i we have: $\sum_{k=1}^{|M_i|}(f(m_k \in M_i) = \tau$.

Using the Jaccard similarity function, two sets M_i and M_j are similar if and only if: $O_s(M_i, M_j) \geq \alpha$ where $\alpha = \frac{t}{1+t}.(|M_i| + |M_j|)$ (Equation (2)). Supposing that the sets were sent to the aggregators without applying the first aggregation phase and without computing measures frequencies, thus we can observe that:

$$|M_i| = |M_j| = \tau \text{ and } f_s(M_i, M_j) = O_s(M_i, M_j). \tag{3}$$

Hence, from Equation (2) and Equation (3) we can deduce that:

$$M_i \text{ and } M_j \text{ are similar iff: } f_s(M_i, M_j) \geq \frac{2 \times t \times \tau}{1 + t}. \tag{4}$$

Frequency filter principle Lemma 1 states that the prefixes of two sets of measures must share at least one measure in order to satisfy the prefix filtering condition (PFC). Nevertheless, in sensor networks this condition is easily satisfied. In this section, we will present an extension of the prefix filtering technique making the PFC condition more difficult to be satisfied.

Lemma 2. *Assume that all the measures in the sets M_i and M_j are ordered according to the global ordering $\mathcal{O}$. Let the p-$prefix$ be the first p elements of M_i. If $f_s(M_i, M_j) \geq \frac{2\times t\times\tau}{1+t}$, then $f_s(p\text{-}M_i, p\text{-}M_j) \geq \sum_{k=1}^{|p\text{-}M_i|}(f(m_k \in p\text{-}M_i)) - \frac{1-t}{1+t} \times \tau$.*

Proof. We denote by $p\text{-}M_i$ the prefix of the set M_i and $r\text{-}M_i$ the set of reminder measures where $M_i = \{p\text{-}M_i + r\text{-}M_i\}$. We have:

$$\begin{aligned} f_s(M_i, M_j) &= f_s(p\text{-}M_i, M_j) + f_s(r\text{-}M_i, M_j) \\ &= f_s(p\text{-}M_i, p\text{-}M_j) + f_s(p\text{-}M_i, r\text{-}M_j) + \\ &\quad f_s(r\text{-}M_i, M_j) \\ &\cong f_s(p\text{-}M_i, p\text{-}M_j) + f_s(r\text{-}M_i, M_j) \\ &\leq f_s(p\text{-}M_i, p\text{-}M_j) + \sum_{k=1}^{|r\text{-}M_i|} (f(m_k \in r\text{-}M_i)) \end{aligned}$$

In the second line we can omit the term $f_s(p\text{-}M_i, r\text{-}M_j)$ because we have assumed that it is negligible compared to the other terms in the equation. Indeed, if the two sets are similar then the measures having highest frequencies must be in the prefix set and not in the reminder, which means that the overlapping between the $p\text{-}M_i$ and $r\text{-}M_j$ is almost empty. From the above equations and equation (4)(similarity condition) we can deduce:

$$\frac{2 \times t \times \tau}{1+t} \leq f_s(p\text{-}M_i, p\text{-}M_j) + \sum_{k=1}^{|r\text{-}M_i|} (f(m_k \in r\text{-}M_i)) \tag{5}$$

From the following equation:

$$\sum_{k=1}^{|p\text{-}M_i|} (f(m_k \in p\text{-}M_i)) + \sum_{k=1}^{|r\text{-}M_i|} (f(m_k \in r\text{-}M_i)) = \tau \tag{6}$$

We obtain:

$$f_s(p\text{-}M_i, p\text{-}M_j) \geq \sum_{k=1}^{|p\text{-}M_i|} (f(m_k \in p\text{-}M_i)) - \frac{1-t}{1+t} \times \tau \tag{7}$$

The lemma is proved.

Algorithm 2 describes our method to find similar sets of measures based on the frequency filtering approach. It is a hybrid solution, where we integrate our frequency condition presented in Lemma 2 to the prefix filtering approach presented in Algorithm 1.

Jaccard similarity computation Although filtering approaches reduce the number of comparisons between the received sets of measures, the number of candidate sets surviving after this phase is still non negligible. Furthermore, the computation of the jaccard similarity between two candidates sets can be very complex, especially when it comes to sensor networks where measures' sets can have ten hundreds or thousands elements. Therefore, to continue filtering out further candidate sets we propose a new frequency filtering constraint in the verification phase. In doing so, we can also reduce the overhead of the jaccard similarity computation.

Algorithm 2 Frequency-filtering based algorithm.

Require: Set of measures' sets $M = \{M_1, M_2...M_n\}$, t, τ.
Ensure: All pairs of sets (M_i, M_j), such that $J(M_i, M_j) \geq t$.

Replace line 5 in Algorithm 1 with

- $Fs \leftarrow$ empty map from set id to int
- $sumFreq \leftarrow 0$
- **for** $k \leftarrow 1$ to p **do**
  $sumFreq \leftarrow sumFreq + f(m_k \in p\text{-}M_i)$
- **end for**

Replace line 10 in Algorithm 1 with

- $Fs[M_j] \leftarrow Fs[M_j] + f_{min}(M_i[k], M_j[l])$

Replace line 18 in Algorithm 1 with

- **for** each M_j such that $Fs[M_j] > sumFreq - \frac{1-t}{1+t} \times \tau$ **do**

Assume that we want to compute the similarity between two sets M_i and M_j. Then, these sets are similar if they satisfy the overlap condition $f_s(M_i, M_j) \geq \frac{2 \times t \times \tau}{1+t}$. We also assume that a measure $m \in M_i$ divides M_i into two partitions: one partition containing all the measures having frequencies higher than $f(m)$ including m denoted by $h\text{-}M_i$ and the second $l\text{-}M_i$ containing all the measures having frequencies less than $f(m)$. Similarly, we assume that any measure in M_j divides it in two partitions $h\text{-}M_j$ and $l\text{-}M_j$. The idea of dividing the sets is to find a measure where at this position a similarity upper bound is estimated and checked against the similarity threshold. As soon as the check is failed we can stop the overlap computing early. This hypothesis is formalized by the following lemma:

Lemma 3. *Assume that $|M_i| < |M_j|$ and all measures in M_i are ordered according to the global ordering $\mathcal{O}$. M_i and M_j are similar $\Rightarrow$ for any $m \in M_i$ dividing M_i into $h\text{-}M_i$ and $l\text{-}M_i$ we have: $f_s(h\text{-}M_i, M_j) \geq \frac{2 \times t \times \tau}{1+t} - \sum_{k=1}^{|l\text{-}M_i|}(f(m_k \in l\text{-}M_i))$.*

Proof. M_i and M_j are similar

$$\Rightarrow f_s(M_i, M_j) \geq \frac{2 \times t \times \tau}{1+t} \tag{8}$$

$$\Rightarrow f_s(h\text{-}M_i, M_j) + f_s(l\text{-}M_i, M_j) \geq \frac{2 \times t \times \tau}{1+t} \tag{9}$$

$$\Rightarrow f_s(h\text{-}M_i, M_j) \geq \frac{2 \times t \times \tau}{1+t} - f_s(l\text{-}M_i, M_j) \tag{10}$$

Then we have:

$$f_s(l\text{-}M_i, M_j) \leq min(\sum_{k=1}^{|l\text{-}M_i|}(f(m_k)), \sum_{k=1}^{|M_j|}(f(m_k))) \quad (11)$$

$$\leq min(\sum_{k=1}^{|l\text{-}M_i|}(f(m_k \in l\text{-}M_i)), \tau) \quad (12)$$

$$\leq \sum_{k=1}^{|l\text{-}M_i|}(f(m_k \in l\text{-}M_i)) \quad (13)$$

From equations (10) and (13) we can deduce that:

$$f_s(h\text{-}M_i, M_j) \geq \frac{2 \times t \times \tau}{1+t} - \sum_{k=1}^{|l\text{-}M_i|}(f(m_k \in l\text{-}M_i)).$$

The lemma is proved.

The algorithm of overlap computation is given in Algorithm 3

Algorithm 3 Overlap Computation.

Require: Two sets of measures M_i and M_j, t, τ.
Ensure: $O_s(M_i, M_j)$.

$O_s \leftarrow 0$
Consider $|M_i| < |M_j|$
$sumFreqH \leftarrow 0$
$sumFreql \leftarrow \tau$
$M_j \leftarrow sort(M_j, |M_j|)$ M_j is sorted in increasing order of the measures
for $k \leftarrow 0$ to $|M_i|$ **do**
    $sumFreql \leftarrow sumFreql - f(M_i[k])$
    Search similar of $M_i[k]$ in M_j
    find $M_j[l]/link(M_i[k], M_j[l]) = 1$
    $sumFreqH \leftarrow sumFreqH + f_{min}(M_i[k], M_j[l])$
    if $sumFreqH \geq \frac{2 \times t \times \tau}{1+t} - sumFreql$ **then**
        $O_s \leftarrow O_s + 1$
    else
        Return $-\infty$
    end if
end for
Return O_s

In this algorithm, we used two kinds of measures ordering depending on the sets sizes. The first one according to the global ordering $\mathcal{O}$ (M_i in the above algorithm) and the second is sorted in increasing order of the measures to accelerate a measure search [1].

[1] in our experiments we used the binary search

6 Experimental Results

To evaluate our approach, we conducted multiple series of simulations using the discrete event simulator OMNET++ [20]. The objective of these simulations is to confirm that our prefix frequency filtering (PFF) technique can successfully achieve desirable results for data aggregation in periodic sensor networks. Therefore, In our simulations we used real readings collected from 45 sensor nodes deployed in the Intel Berkeley Research Lab [21]. Every 31 seconds, sensors with weather boards were collecting humidity, temperature, light and voltage values. For the sake of simplicity, in this paper we are interested in one field of sensor measurements: the temperature [2]. We performed several runs of the algorithms (an average of 15 runs). In each experimental run, we generated a network of 46 nodes corresponding to those was deployed in the Intel Berkeley Lab. Each node then reads periodically real measures saved in a file while applying the first aggregation algorithm. At the end of this step, each node sends its set of measures/frequencies to an aggregator node which in his turn applies prefix and filtering algorithms to theses sets. Furthermore, we compare our approach to the ToD protocol proposed in [13] [14]. As our real data sensor network consists of 46 nodes, we use ToD in a one dimensional Network as explained in [14] and we only divide the network into two F-cluster.

We evaluated the performance of the protocols using the following parameters: **a)** the number of sensor measurements taken by all nodes during a period τ, and **b)** the threshold of the Jaccard similarity function t. The threshold δ is fixed to 0.07. The aggregation function used for the ToD protocol is the same used in our approach (PFF) based on the link function (cf section 4). We employ four metrics in our simulations:

- The number of candidate sets generated after applying the prefix filtering approach [3], the frequency filtering algorithms with optimizations (PFF) and the final result (the real number of duplicate sets);
- Percentage of received measures: It represents how effective a protocol is in aggregating data. It is the number of measures received by the sink over the number of measures taken by all nodes.
- Data accuracy: represents the measures loss rate. It is a evaluate of measures taken by the source nodes and did not received at the base station (sink). It is defined also as the aggregation error.
- Overall energy dissipation: is the total energy dissipation of the entire network. To evaluate the energy consumption of our approach we used the same radio model as discussed in [21].

6.1 Prefix frequency filtering optimizations

In this section we compared the number of candidates (number of comparisons) generated respectively by our frequency filtering technique (PFF), the prefix filtering algorithm and the results obtained after applying the Jaccard similarity function. We

[2] the others are done by the same manner

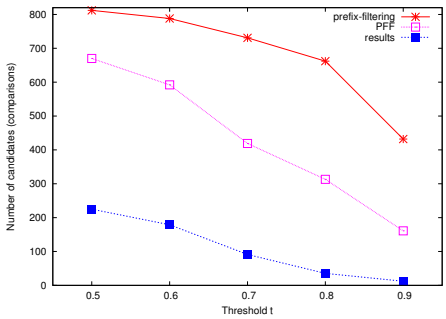

Fig. 1. Sets comparison

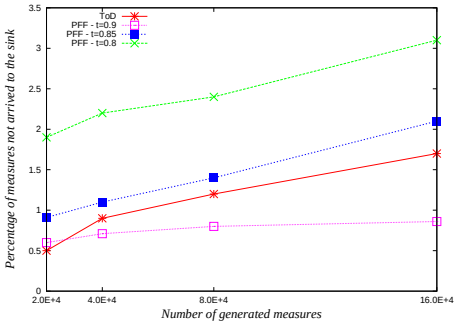

Fig. 3. Data accuracy

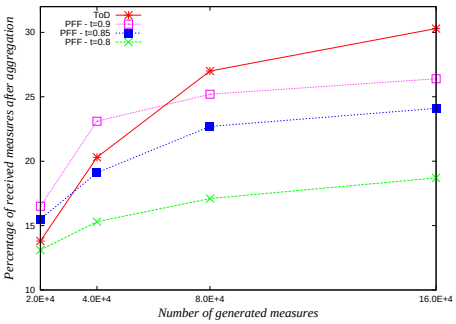

Fig. 2. Received measures

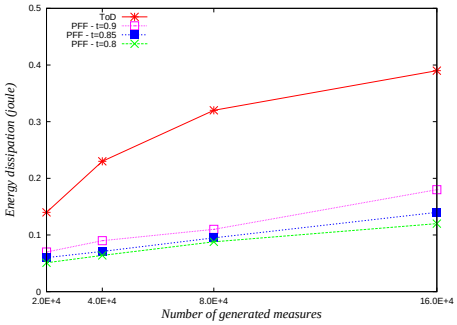

Fig. 4. Total energy dissipation

fixed the number of the total measuremtns taken by all the nodes during a period to $\tau = 8.E + 04$. The obtained result is shown in figure 1. We notice that, when the similarity threshold increases from 0.7 to 0.9, the number of comparisons of the frequency filtering and the prefix filtering becomes closer. We can also see that our frequency filtering technique (PFF) outperforms the prefix filtering methods in all cases. Moreover, the number of candidates generated by all the algorithms is far bigger than the results number. This is to prove that under this circumstance, applying early termination algorithm is very effective (Algorithm 3).

6.2 Percentage of received measures and data accuracy

Figure 2 shows the percentage of received measures over the total number taken by all nodes for the temperature field. These experiments permit to show how well aggregation protocols do aggregation and reduce redundant measures. PFF performs better than ToD in terms of data aggregation because of it is ability to compare sets of data instead of single packets. In other words, PFF reduces the number of redundant data traveling into the networks better than TOD especially when the number of readings increase (the case of periodic networks). We also notice that, the percentage of received packets remains almost unchangeable while increasing the sensor readings.

Figure 3 depicts the resulsts of the aggregation error. This metric is an important performance index, and the high measures loss rate will impact the use of the data greatly. The obtained results show that the two protocols have good performance regarding the aggregation error. As expected, when we increase the threshold t of the similarity function we reduce the measures loss rate. For instance, we can notice that PFF outperforms ToD in terms of data accuracy for $t = 0.9$.

6.3 Overall energy dissipation

The overall energy dissipation is the total energy consumption of the entire network. Figure 4 shows the results for total energy consumption obtained while varying the total number of sensor readings. The figure shows that the overall energy dissipation for different protocols increases as the number of readings increases. We notice that ToD consumes not too much, but does not scale well as the number of readings increases. For all the values of the threshold t tested, PFF always outperforms the ToD protocol in total energy dissipation. This is because, the packet-packet comparison used in ToD instead of data sets in PFF generates more transmissions in the network, furthermore, the packet construction in ToD contains additional information required for the aggregation which is not the case in PFF.

7 Conclusion and future work

In this paper we proposed a tree based bi-level model for data aggregation in periodic sensor networks: Local aggregation and Frequency filtering aggregation. In the first one we provided an aggregator for simple captured measurements based on a link similarity function while in the second level our objective is to detect and aggregate multiple data sets generated by different neighboring nodes. We proposed a new frequency filtering approach and several optimizations using sets similarity functions to find similar data sets. It was shown through simulations on real data measurements that our method reduces drastically the redundant sensor measures and outperforms the existing prefix filtering approaches.

We have two major directions for our future work. The first direction seeks to adapt our proposed method to take into account reactive periodic sensor networks, where sensor nodes operate with different sampling rate. In periodic applications the dynamics of the monitored condition or process can slow down or speed up; and to save more energy the sensor node can adapt its sampling rates to the changing dynamics of the condition or process. The second direction is to develop a new suffix frequency filter algorithm beside the frequency filtering approach proposed in this paper. Our goal is to use additional filtering method that prunes erroneous candidates that survive after applying the prefix and frequency filtering technique.

References

1. Bo Yu, Jianzhong Li, and Yingshu Li. Distributed data aggregation scheduling in wireless sensor networks. *IEEE, INFOCOM2009*, 2009.

2. Yanfei Zheng, Kefei Chen, and Weidong Qiu. Building representative-based data aggregation tree in wireless sensor networks. *Mathematical Problems in Engineering*, page 11 pages, 2010.
3. Jacques Bahi, Abdallah Makhoul, and Maguy Medlej. Data aggregation for periodic sensor networks using sets similarity functions. *IWCMC 2011, 7th IEEE Int. Wireless Communications and Mobile Computing Conference*, pages 559–564, July 2011.
4. M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis. Tina: A scheme for temporal coherency-aware in-network aggregation. *3rd ACM international workshop on Data engineering for wireless and mobile access*, pages 69–76, 2003.
5. Yingqi Xu, Wang-Chien Lee, Jianliang Xu, and G. Mitchell. Processing window queries in wireless sensor networks. *22nd Int. Conf. on Data Engineering. ICDE*, page 70, 2006.
6. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: A tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev*, 36(SI):131–146, 2002.
7. G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. *2005 ACM SIGMOD International Conference on Management of Data*, pages 25–36, 2005.
8. G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Prolonging the lifetime of wireless sensor networks via unequal clustering. *Proceedings of the 5th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks*, 2005.
9. SangHak Lee and TaeChoong Chung. Data aggregation for wireless sensor networks using self-organizing map. *Artificial Intelligence and Simulation Lecture Notes in Computer Science*, pages 508–517, 2005.
10. Huifang Chen, Hiroshi Mineno, and Tadanori Mizuno. Adaptive data aggregation scheme in clustered wireless sensor networks. *Computer Communications*, 31(15):3579–3585, 2009.
11. R.C. Shah and J.M Rabaey. Energy aware routing for low energy ad hoc sensor networks. *IEEE Wireless Communications and Networking Conf. WCNC*, pages 350–355, 2002.
12. O. Younis and S. Fahmy. An experimental study of routing and data aggregation in sensor networks. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, page 8 pages, 2005.
13. Prakash G L, Thejaswini M, S H Manjula, K R Venugopal, and L M Patnaik. Tree-on-dag for data aggregation in sensor networks. *World Academy of Science, Engineering and Technology*, 37, 2009.
14. Kai-Wei Fan, Sha Liu, and Prasun Sinha. Dynamic forwarding over tree-on-dag for scalable data aggregation in sensor networks. *IEEE Trans. on Mobile Computing*, 7(10):1271–1284, 2008.
15. Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. Scaling up all pairs similarity search. *16th international conference on World Wide Web, WWW'07*, pages 131–140, 2007.
16. Sunita Sarawag and Alok Kirpal. Efficient exact set-similarity joins. *32nd international conference on Very large data bases, VLDB'06*, pages 918–929, 2006.
17. Surajit Chaudhuri, Venkatesh Ganti, and Raghav Kaushik. A primitive operator for similarity joins in data cleaning. *22nd International Conference on Data Engineering (ICDE'06)*, page 5, 2006.
18. Chuan Xiao, Wei Wang, Xuemin Lin, and Jeffrey Xu Yu. Efficient similarity joins for near duplicate detection. *Proceeding of the 17th international conference on World Wide Web*, pages 131–140, ACM 2008.
19. Chuan Xiao, Wei Wang, Xuemin Lin, and Haichuan Shang. Top-k set similarity joins. *Proceedings of the 2009 IEEE International Conference on Data Engineering*, pages 916–927, 2009.
20. OMNeT++. http://www.omnetpp.org/.
21. Samuel Madden. http://db.csail.mit.edu/labdata/labdata.html.